BRIEF DESCRIPTION OF THE DRAWINGS

[0011] For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying Drawings in which:

KB 8/18/04 [0012] FIGURE ~~1~~ 1a illustrates an overall block diagram of the system for both preprocessing data during the training mode and for preprocessing data during the run mode;

KB 8/18/04 [0013] FIGURE ~~1a~~ 1b illustrates a simplified block diagram of the system of FIGURE ~~1~~ 1a;

[0014] FIGURE 2 illustrates a detailed block diagram of the preprocessor in the training mode;

[0015] FIGURE 3 illustrates a simplified block diagram of the time merging operation, which is part of the preprocessing operation;

[0016] FIGUREs 4a and 4b illustrate data blocks of the before and after time merging operation;

[0017] FIGUREs 5a-5c illustrate a diagrammatic view of the time merging operation;

[0018] FIGURE 6 illustrates a flowchart depicting the preprocessing operation;

[0019] FIGUREs 7a-7f illustrate the use of graphical tools for cleaning up the "raw" data;

[0020] FIGURE 8 illustrates the display for the algorithm selection operation;

[0021] FIGURE 9 illustrates a block diagram of a plan depicting the various places in the process flow that parameters occur relative to the plant output;

[0022] FIGURE 10 illustrates a diagrammatic view of the relationship between the various plant parameters and the plant output;

[0023] FIGURE 11 illustrates a diagrammatic view of the delay provided for input data patterns;

DETAILED DESCRIPTION OF THE INVENTION

KB 8/18/04 **[0044]** Referring now to FIGURE ~~1~~ 1a, there is illustrated an overall block diagram of the data preprocessing operation in both the training mode and the run-time mode. In the training mode, one of more data files 10 are provided, which data files include both input training data and output training data. The training data is arranged in "sets", which sets correspond to different plant variables, and which may be sampled at different time intervals. This data is referred to as the "raw" data. When the data is initially presented to an operator, the data is typically unformatted, i.e., each set of data is in the form that it was originally received. Although not shown, the operator will first format the data files so that all of the data files can be merged into a data-table or spreadsheet, keeping track of the original "raw" time information. This is done in such a manner as to keep track of the time stamp for each variable. Thus, the "raw" data is organized as time, value pairs of columns; that is, for each variable x_i, there is its associated time of sample t_i. The data can then be grouped into sets $\{x_i, t_i\}$.

[0045] If any of the time-vectors happen to be identical, it is convenient to arrange the data such that the data will be grouped in common time scale groups, and data that is on, for example, a fifteen minute sample time scale will be grouped together and data sampled on a one hour sample time scale will be grouped together. However, any type of format that provides viewing of multiple sets of data is acceptable.

[0046] The data is input to a preprocessor 12 that functions to perform various preprocessing functions, such as reading bad data, reconciling data to fill in bad or missing data, and performing various algorithmic or logic functions on the data. Additionally, the preprocessor 12 is operable to perform a time merging operation, as will be described hereinbelow. During operation, the preprocessor 12 is operable to store various preprocessing algorithms in a given sequence in a storage area 14. As will

be described hereinbelow, the sequence defines the way in which the data is manipulated in order to provide the overall preprocessing operation.

[0047] After preprocessing by the preprocessor 12, the preprocessed data is input to a delay block 16, the delay block 16 operable to set the various delays for different sets of data. This operation can be performed on both the target output data and the input training data. The delay settings are stored in a storage area 18 after determination thereof.

[0048] The output of the delay block 16 is input to a training model 20. The training model 20 is a non-linear model that receives input data and compares it with target output data and trains the network to generate a model for predicting the target output data from the input data. In the preferred embodiment, the training model utilizes a multi-layered neural network that is trained on one of multiple methods, one being Back Propagation. Various weights within the network are set during the Back Propagation training operation, and these are stored as model parameters in a storage area 22. The training operation and the neural network are conventional systems.

[0049] A Distributed Control System (DCS) 24 is provided that is operable to generate various system measurements and control settings representing system variables such as temperature, flow rates, etc., that comprise the input data to a system model. The system model can either generate control inputs for control of the DCS 24 or it can provide a predicted output, these being conventional operations. This is provided by a run-time system model 26, which has an output 28 and an input 30. The input 30 is comprised of the preprocessed and delayed data and the output can either be a predictive output, or a control input to the DCS 24. In the embodiment of FIGURE ~~1~~ 1a this is illustrated as control inputs to the DCS 24. The run-time system model 26 is utilizing the model parameters stored in the storage area 22. It should be noted that the run-time system model 26 contains a representation learned during the training

KO 8/18/04

operation, which representation was learned on the preprocessed data. Therefore, data generated by the DCS 24 must be preprocessed in order to correlate with the representation stored in the run-time system model 26.

[0050] The DCS 24 has the data output thereof input to a run-time preprocess block 34, which is operable to process the data in accordance with the sequence of preprocessing algorithms stored in the storage area 14, which were generated during the training operation. The output of the run-time preprocessor 34 is input to a run-time delay box 36 to set delays on the data in accordance with the delay settings stored in the storage area 18. This provides the overall preprocessed data output on the line 34 input to the run-time system model 26.

LB 8/18/04 [0051] Referring now to FIGURE ~~1a~~ 1b, there is illustrated a simplified block diagram of the system of FIGURE ~~1~~ 1a, wherein a single preprocessor 34' and a single delay 36' are utilized. The output of the delay 36' is input to a single system model 26'. In operation, the preprocessor 34', the delay 36' and the system model 26' operate in both a training mode and a run-time mode. A multiplexer 35 is provided that receives the output from the data file(s) 10 and the output of the DCS 24, this providing plant variables of the DCS 24, the output of the multiplexer input to the preprocessor 34'. A control device 37 is provided that controls the multiplexer 35 to select either a training mode or a run-time mode. In the training mode, the data file(s) 10 has the output thereof selected by a multiplexer and the preprocessor 34' is operable to preprocess the data in accordance with a training mode, i.e., the preprocessor 34' is utilized to determine what the predetermined algorithm sequence is that is stored in the storage area 14. An input/output device I/O 41 is provided for allowing the operator to interface with the control device 37. The delay 36' is also controlled by the control device 37 to determine the delay settings for storage in the storage area 18. The system model 26' is operated in a training mode such that the target data and the input data to the system model 26' are generated, the training controlled by training block 39. The training block 39 is